

CLAIMS

What is claimed is:

1. A chain of snapshots for preserving data, comprising:
 - a first snapshot, wherein the first snapshot is a read only snapshot that cannot be written by a user; and
 - a second snapshot descending from the first snapshot, wherein the second snapshot is a read-write snapshot that can be written by the user.
2. The chain of snapshots of claim 1, further comprising a third snapshot descending from the first snapshot, wherein the third snapshot is another read-write snapshot.
3. The chain of snapshots of claim 1, further comprising a third snapshot descending from the second snapshot, wherein the third snapshot is another read only snapshot.
4. The chain of snapshots of claim 1, wherein at least one of the first and the second snapshots comprises:
 - a table comprising a plurality of entries corresponding to first parts of block addresses;
 - wherein one of the plurality of entries contains a pointer to contiguous data blocks in said at least one of the first and the second snapshots.
5. The chain of snapshots of claim 1, wherein at least one of the first and the second snapshots comprises:
 - a first table comprising a first plurality of entries corresponding to first parts of block addresses;
 - a second table comprising a second plurality of entries corresponding to second parts of block addresses;

wherein one of the first plurality of entries contains a pointer to the second table and one of the second plurality of entries contains a pointer to contiguous data blocks in said at least one of the first and the second snapshots.

6. A method for writing to data blocks in snapshots for preserving data, comprising:

creating a first snapshot, wherein the first snapshot is a read only snapshot that cannot be written by a user; and

creating a second snapshot descending from the first snapshot, wherein the second snapshot is a read-write snapshot that can be written by the user.

7. The method of claim 6, further comprising creating a third snapshot descending from the first snapshot, wherein the third snapshot is another read-write snapshot.

8. The method of claim 6, further comprising creating a third snapshot descending from the second snapshot, wherein the third snapshot is another read only snapshot.

9. The method of claim 6, wherein at least one of said creating a first snapshot and said creating a second snapshot comprises:

creating a first table comprising a first plurality of entries corresponding to first parts of block addresses.

10. The method of claim 9, further comprising writing one or more data blocks to the first snapshot or the second snapshot, said writing comprising:

saving said one or more data blocks in the page of contiguous data blocks; and

writing one of the first plurality of entries with a pointer to the contiguous data blocks if said one or more data blocks consist all of the data blocks with the same first part of their block addresses.

11. The method of claim 9, further comprising writing one or more data blocks to the first snapshot or the second snapshot, said writing comprising:

creating a second table comprising a second plurality of entries corresponding to second parts of block addresses;

saving said one or more data blocks in a page of contiguous data blocks;

writing one of the first plurality of entries with a pointer to the second table; and

writing one of the second plurality of entries with a pointer to the contiguous data block if said one or more data blocks consist all of the data blocks with the same first part and the same second part of their block addresses.

12. The method of claim 9, further comprising, prior to said writing the data block:

determining if the write request is a first write to the data block in the second snapshot; and

if the write request is the first write to the data block in the second snapshot, copying the data block to a third snapshot descending from the second snapshot, wherein the third snapshot is a read-only snapshot.

13. A method for reading a data block having a block address from a snapshot, comprising:

using a first part of the block address to look up one of a first plurality of entries in a first table in the snapshot;

if said one of the first plurality of entries contains a pointer to a first page of contiguous data blocks in the snapshot, following the pointer to the first page and reading the first page; and

if said one of the first plurality of entries contains a pointer to a second table in the snapshot, following the pointer to the second table.

14. The method of claim 13, further comprising, if said one of the first plurality of entries contains a pointer to a second table in the snapshot:

using a second part of the block address to look up one of a second plurality of entries in the second table;

if said one of the second plurality of entries contains a pointer to a second page of contiguous data block in the snapshot, following the pointer to the second page and reading the second page; and

if said one of the second plurality of entries contains a pointer to a third table in the snapshot, following the pointer to the third table.

15. The method of claim 14, further comprising, if said one of the second plurality of entries contains a pointer to a third table in the snapshot:

using a third part of the block address to index one of a third plurality of entries in the third table;

if said one of the third plurality of entries contains a pointer to the data block in the snapshot, following the pointer to the data block and reading the data block.

16. A chain of snapshots, comprising:

a more recent snapshot comprising a data block copied from a base volume;

a less recent snapshot comprising a skip-pointer that leads to the data block in said more recent snapshot; and

one or more intermediate snapshots between said more recent snapshot and said less recent snapshot in the chain of snapshots.

17. The chain of snapshots of claim 16, wherein the skip-pointer leads to said more recent snapshot.

18. The chain of snapshots of claim 16, wherein the skip-pointer leads to an address table in said more recent snapshot and the address table at least partially represents the address of the data block.

19. The chain of snapshots of claim 16, wherein the skip-pointer directly leads to the data block in said more recent snapshot.
20. The chain of snapshots of claim 16, wherein said one or more intermediate snapshots comprising the skip-pointer to the more recent snapshot.
21. A method for generating skip-pointers in snapshots, comprising:
 - generating a chain of snapshots descending from a parent;
 - receiving a write request to a data block in the parent;
 - determining if the write is a first to the data block in the parent;
 - if the write is the first to the data block in the parent:
 - copying the data block to a most recent snapshot in the chain of snapshots;
 - writing a skip-pointer in at least one of the other snapshots in the chain of snapshots, the skip-pointer leading to the most recent snapshot;
 - writing the data block in the parent;
 - if the write is not the first to the data block, only writing the data block in the parent.
22. The method of claim 21, wherein the parent is a base volume.
23. The method of claim 21, wherein the parent is a read-write snapshot that can be written by a user.
24. The method of claim 21, wherein the skip-pointer leads to address tables in the most recent snapshot.
25. The method of claim 21, wherein the skip-pointer leads to the data block in the most recent snapshot.

26. The method of claim 21, further comprising reading the data block from said at least one of the other snapshots, comprising:

searching for the data block in said at least one of the other snapshots;

finding the skip-pointer in said at least one of the other snapshots;

following the skip-pointer to the data block; and

reading the data block.

27. A method for generating skip-pointers in snapshots, comprising:

searching for a data block in a selected snapshot in a chain of snapshots;

if the data block is not found in the selected snapshot, searching for the data block in more recent snapshots in the chain of snapshots;

if the data block is found in one of the more recent snapshots, writing a skip-pointer in the selected snapshot that leads to the data block; and

reading the data block in said one of the more recent snapshots.

28. The method of claim 27, wherein the skip-pointer leads to said one of the more recent snapshots.

29. The method of claim 27, wherein the skip-pointer leads to an address table in said one of the more recent snapshots and the address table at least partially represents the address of the data block.

30. The method of claim 27, wherein the skip-pointer leads to the data block in said one of the more recent snapshots.

31. The method of claim 27, further comprising:

searching again for the data block in the selected snapshot;

finding the skip-pointer in the selected snapshot;

following the skip-pointer; and

reading the data block in said one of the more recent snapshots.

32. The method of claim 27, further comprising, if the data block is found in one of the more recent snapshots:

writing the skip-pointer in at least one of intermediate snapshots in the chain of snapshots between the selected snapshot and said one of the more recent snapshots.

33. The method of claim 32, further comprising:

searching for the data block in said one of the intermediate snapshots;

finding the skip-pointer in said one of the intermediate snapshots;

following the skip-pointer; and

reading the data block in said one of the more recent snapshots.

34. The method of claim 27, further comprising, if the data block is not found in one of the more recent snapshots:

writing the skip-pointer in the selected snapshot, the skip-pointer leading to a most recent snapshot in the chain of snapshots.

35. The method of claim 34, further comprising:

searching again for the data block in the selected snapshot;

finding the skip-pointer in the selected snapshot;

following the skip-pointer; and

searching for the data block in the most recent snapshot.

36. A method for deleting a snapshot in a chain of snapshots, comprising:
- selecting a first data block in said snapshot, said first data block having a block address;
 - searching for a second data block with said block address in an adjacent less recent snapshot;
 - if said second data block is not found in said adjacent less recent snapshot, copying said first data block from said snapshot to said adjacent older snapshot;
 - deleting said snapshot.
37. The method of claim 36, further comprising:
- searching for a skip-pointer in another less recent snapshots in the chain, said skip-pointer leading to said first data block in said snapshot; and
 - updating said skip-pointer to lead to said first data block in said adjacent less recent snapshot.
38. A method for deleting a snapshot from a chain of snapshots, comprising:
- selecting a first data block in the snapshot, the first data block having a block address;
 - searching for a second data block with said block address in an adjacent less recent snapshot;
 - if the second data block is not found in the adjacent less recent snapshot:
 - writing a pointer corresponding to said block address in the adjacent less recent snapshot, wherein the pointer leads to the first data block;
 - deleting the snapshot but for the first data block.